Network Lab Setup and Documentation

1. Overview

This network lab simulates a small virtual network designed for learning, troubleshooting, and experimentation.

Components:

- **Router VM:** Provides NAT, DHCP, firewall rules, and controls network traffic.
- **Client VM (Ubuntu):** General-purpose client that accesses services hosted by the server.
- **Server VM (Ubuntu):** Runs an Apache web server; equipped with self-recovery logic.
- **Monitor VM (Kali Linux):** Dedicated for packet capture and traffic analysis.

---

2. Environment

- All VMs are connected through a virtual network (e.g., Internal Network or Virtual Switch).
- IP addresses are dynamically assigned via DHCP from the router.
- Promiscuous mode is enabled on the Monitor VM to capture all traffic.
- Kali Linux is used for monitoring and analysis tools.

---

3. Network Configuration

DHCP & NAT:

- DHCP service is active on the router VM.

- NAT is configured to allow client/server VMs internet access via the router.

Firewall:

- **iptables** firewall configured on router:
  - Default policy: DROP all forwarded traffic.
  - **Allowed:** Client → Server on ports 80 (HTTP) and 22 (SSH).
  - All other inter-VM traffic is denied.

```
sudo iptables -A FORWARD -s 192.168.1.10 -d 192.168.1.20 -p tcp --dport 80 -j ACCEPT
sudo iptables -A FORWARD -s 192.168.1.10 -d 192.168.1.20 -p tcp --dport 22 -j ACCEPT
sudo iptables -P FORWARD DROP
```

---

4. Software and Services

Server VM:

- **Apache Web Server** installed (`apt install apache2`).
- Hosting custom HTML content at `/var/www/html/index.html`.
- Self-healing watchdog script monitors gateway and restarts the interface if needed.

Client VM:

- Can access server over HTTP.
- Runs its own watchdog script to recover from interface failures.

Monitor VM:

- Runs **Wireshark** and **tcpdump**.

- Configured with **promiscuous mode** on interface.
- Captures and inspects traffic between client and server.

---

5. Security

- Traffic is tightly controlled via router firewall.
- Server access is limited to only the client.
- Monitor VM is passive and not routable by design.
- Watchdog scripts on client and server provide recovery, reducing downtime from network issues.

---

6. Failure Simulation & Auto-Recovery

Network Failure Simulation:

- Bring interfaces down using `ip link set eth0 down`.
- Block traffic via iptables rules.
- Disconnect virtual network adapters.

Auto-Recovery Implementation:

- **Client & Server VMs** run custom watchdog scripts as `systemd` timer units.
- Scripts ping a known IP (router or server), and restart the interface if the host is unreachable.
- Logs are stored at `/var/log/network-watchdog.log` and `/var/log/server-watchdog.log`.

Example Client Script (Runs Every Minute):

```
ping -c 3 192.168.1.20 || {
  ip link set eth0 down
  sleep 5
  ip link set eth0 up
}
```

---

7. Troubleshooting and Learning Outcomes

- Learned to deploy web services and restrict access via iptables.
- Understood watchdog and `systemd` timers for automated recovery.
- Practiced capturing and analyzing traffic with Wireshark/tcpdump.
- Built a multi-tiered lab environment suitable for offensive and defensive testing.
